Object-Oriented Programming I (CSE 121)
Week # 09
(Abstraction in C++)

Objective(s)

In this lecture, you will learn:

1. What is Abstraction?
2. Real-life examples of Abstraction.
3. How abstraction is achieved in a class?
4. Abstraction example

Abstraction in C++

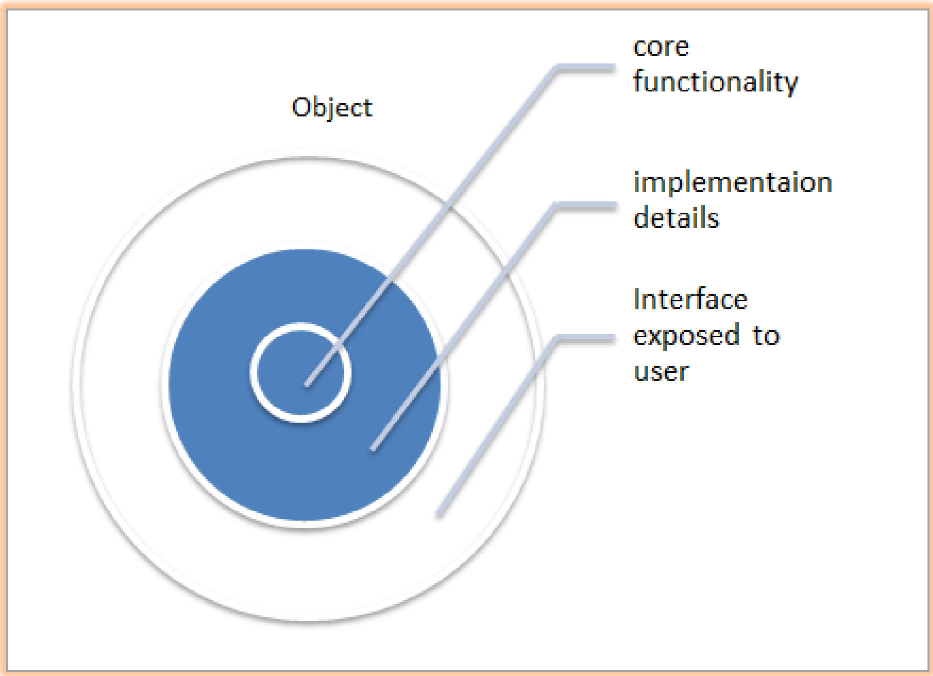
Object-Oriented Programming (OOP) has four core concepts which are listed below:

- Encapsulation
- Abstraction
- Inheritance
- Polymorphism

In this lecture, we will explore all about data abstraction in C++.

Abstraction

- **Abstraction** is a technique of object-oriented programming which help us to hide the details of implementation from the user and only expose an interface that is required.
- **Abstraction** is one of the four pillars of object-oriented programming and almost all the OOP solutions are based on the abstraction principles i.e. separation of interface and implementation details in the program.



- In the above diagram, we have shown an object and its contents as a pictorial representation. We can see that the innermost layer of this object is its core functionality followed by the implementation details.
- In OOP, these two layers (though in most of the cases this is just one layer) are not exposed to the outside world. The outermost layer, the interface, is the layer that is provided for the end user in order to access the functionality of the object.
- As a result, any changes made to the innermost layers of the object are not evident to an end user as long as the interface that the user is exposed to remains the same.

Real life example of Abstraction

Abstraction shows only important things to the user and hides the internal details.

- **Example 1:** Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc. in the car.



- **Example 2:** Suppose you are operating a mobile phone. You know that the important feature of a phone is the memory card, SIM, battery life, design and build, and processor power. But while operating the phone you do not get into the operational details of the phone such as how the CPU of the phone allocates memory for the various media on your phone or other intricate architectural details of the phone. All these details aren't visible to cell-phone users from a non-technical background.



- **Example 3:** When you send an email to someone you just click send and you get the success message, what actually happens when you click send, how data is transmitted over network to the recipient is hidden from you (because it is irrelevant to you).
- **Example 4:** ATM Machine is an example of abstraction; All are performing operations on the ATM machine like cash withdrawal, money transfer, retrieve mini-statement etc. but we can't know internal details about ATM.
- **Example 5:** When we using our laptop, we only know how to use it, but we don't know exactly how it's work. On another side, when our laptop is locked, the user can't see our personal information on the laptop. So, when he will put the password and unlock the window then he'll be able to access the information. So, this window password work like abstraction.

How Abstraction is achieved in a class

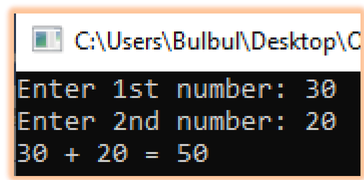
A class implemented in C++ with access specifiers public, private and protected, can be considered as an implementation of abstraction.

Abstraction Example in C++

Example 1: A simple example of Abstraction.

```cpp
//Abstraction implemented using class and access modifiers.
#include <iostream>
using namespace std;

class  Sum{
private:
    int n1, n2; //private data members (Abstraction)
    //private member function (Abstraction)
    void setValue(){
        cout << "Enter 1st number: ";
        cin >> n1;
        cout << "Enter 2nd number: ";
        cin >> n2;
    }
public:
    void getSum(){
        setValue();
        cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    }
};
int main(){
    Sum s;
    s.getSum();
  return 0;
}
```



Example explained

- In the above program, we have a class named '**Sum**' which has two integer variables, n1 and n2. It also has two functions setValue() and getSum(). The data members' n1 and n2, as well as the function setValue(), are private to the class.
- The function getSum() is public to the class. In the main function, we create an object of class Sum and call getSum() that reads the two numbers as well as prints their sum.
- This is the abstraction implementation. We expose only one function to the public while keeping the other data members and functions under wraps.
- Though this is just an example to demonstrate abstraction, while implementing real-life problems, we can have many levels of abstraction in C++.

Conclusion: Abstraction is one of the most important concepts in OOP and is implemented at a great depth in C++. Using abstraction, we can keep the implementation details of the program under wraps and only expose the details that we want to the outside world.

……………………………